Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 46 (2012) 1522 – 1526

Procedia Social and Behavioral Sciences

WCES 2012

Synchronous partial replication – case study: implementing e-learning platform in an academic environment

Ciobanu (Iacob) Nicoleta - Magdalena [a] * , Ciobanu (Defta) Costinela Luminiţa [b]

[a] *Ph.D. student, Faculty of Mathematics and Computer Science /Department of Computer Science, University of Pitesti, Targu din Vale Street, No.1, Arges and Postcode 110040, Romania*

[b] *Faculty of Touristic and Commercial Management, "Dimitrie Cantemir" Christian University, Unirii Blvd, No 176, District 4, Bucharest, Romania*

Abstract

Replication is the key factor in improving availability, fault-tolerance and accessibility of data in all types of distributed systems. Replicated data is stored at multiple sites so that it can be used even when some copies are not available due to site failures and databases are kept synchronized to maintain consistency. The scope of distributing these replicas is local data processing. This paper describes the importance of synchronous partial replication, implementation and query optimization steps using Oracle technology for an E-learning portal of a university with geographically distributed locations.

Keywords: Distributed databases, replication, methods, strategies;

1. Introduction

The distributed databases are a qualitative leap in databases evolution and development, opening new perspectives in the design and implementation of information systems. A distributed database system consists of a collection of local databases, geographically located in different points (nodes of a network of computers) and logically related by functional relations so that they can be viewed globally as a single database.

In centralized databases all data are placed on a single node, only users being distributed in the network, and are managed by a single DBMS. In this case, the disadvantages are especially high costs of communication and a very low reliability and availability because any error that blocks access to the database break all activity on the network. Distributed databases eliminate disadvantages of centralized databases and offer several advantages such as availability, reliability, performance, modular development etc. But in distributed environments we face new problems that are not relevant in centralized environments, such as *fragmentation* and *data replication*.

A data fragment constitutes some subset of the original database. A data replica constitutes some copy of the whole or part of the original database. The fragmentation and the replication can be combined: a relationship can be

*Ciobanu (Iacob) Nicoleta Magdalena. Tel.: +40-757-011-895 .

E-mail addresses: nicoleta.iacob_2007@yahoo.com; lumi.defta@yahoo.com .

This work was partially supported by the strategic grant POSDRU/88/1.5/S/52826, Project ID52826 (2009), co-financed by the European Social Fund – Investing in People, within the Sectoral Operational Programme Human Resources Development 2007-2013 .

 doi:10.1016/j.sbspro.2012.05.333

partitioned into several pieces and can have multiple replicas of each fragment (Silberschatz, Korth & Sudarshan, 2010).

Definition 1.1. A **distributed database system** consists of a collection of sites, connected together by a communications network, in which:

- each site is a full database system site in its own right, but
- the sites have agreed to work together so that a user at any site can access data anywhere in the network exactly as if the data were all stored at the user's own site.

Definition 1.2. It follows that a **distributed database (DDB)** is really a kind of virtual database, whose component parts are physically stored in a number of distinct "real" databases at a number of distinct sites (in effect, it is the logical union of those database) (Date, 2003).

Definition 1.3. A **distributed database management system (DDBMS)** is a software system that manages a distributed database while making the distribution transparent to the user (Ozsu & Valduriez, 2011).

Definition 1.4. Let $K_i = \{ A_{i1}, \ldots, A_{ik} \}$, k = 1, 2, … be a *collection of data,* which consists of a set of attributes (fields, columns) and has an associated set of data (lines, tuples, records). The **database**: **BD** = $\{ K_{1, \ldots, K_{i,\ldots}} \}$, $i \in I$, I = 1, 2,…,n is a collection of structured data (Iacob, 2011). A database is simply a collection of data stored on a computer.

Definition 1.5. Let $C = \{ C_{1, \ldots, C_j} \}$, $j \in J$, J = 1, 2,…,m be a set of computers, then **DDB** = $\{ P_1 \cup P_2 \cup \ldots \cup P_j \}$ is a **distributed database** and $P_1, P_2, \ldots, P_j$ the parties that compose it, where $\{ P_1 \cap \ldots \cap P_j \} = \emptyset$, P_j is a corresponding part of the computer C_j, $P_j \in \{K_{1, \ldots, K_{i,\ldots}}\}$, $i \in I$, with $2 \leq j \leq m$, i.e. P_j is a subset of the DDB database. The term distributed simply means that more than one computer might cooperate in order to perform some tasks.

2. Data replication

Definition 2.1. Replication is the operation of storing portions from a database, as copies, on multiple nodes on a network. If a user updates a local copy then DDBMS automatically updates all copies of that data. Assuming replicas are mutually consistent, replication improves availability since a transaction can read any of the copies. In addition, replication provides more reliability, minimizes the chance of total data loss, and greatly improves disaster recovery. Although replication gives to the system better read performance, it does affect the system negatively when database copies are modified. That is because an update operation, for example, must be applied to all of the copies to maintain the mutual consistency of the replicated items (Rahimi & Haug, 2010).

By storing the data at more than one site, if a data site fails, a system can operate using replicated data, thus increasing availability and fault tolerance. At the same time, as the data is stored at multiple sites, the request can find the data closer to the site where the request originated, thus increasing the performance of the system.

In distributed DBMS it is assumed that a replicated database should behave like a database system managing a single copy of the data. As replication is transparent from users' point of view, users may want to execute interleaved executions on a replicated database in a way equivalent to a *one-copy* database, the criterion commonly known as *one-copy serializability* (1SR).

Although data replication has clear benefits, it poses the considerable challenge of keeping different copies synchronized.

In **synchronous replication**, replicas are kept in sync at all times. Synchronous replication means that an update should be propagated and applied immediately in all replicas, so this way database consistency is assured all the time. In this approach, a transaction may access any copy of data because the accessed data are the same in all replicas from different locations. Some applications, for example financial or places reservation, are well served by this technique. This process is usually done using a protocol called "two phase commit" and refers to the transaction that performs the update. Synchronous system updates all the replicas before the transaction commits.

In a synchronous replication scheme, all data is committed to both sets of storage before the client that wrote the data is informed the data has been written.

Partial Replication: While full replication places copies of data items at all replicas, partial replication only assigns copies of an individual data item to some replicas. When there is a high update workload full replication has too much overhead to keep all copies consistent and the individual replicas have little resources left to execute read operations. In contrast, with partial replication a replica has to execute only the updates for data items for which it has local copies, and thus, has more potential to execute read operations. The characteristics of this method are: increased redundancy, decreased concurrency access to data, medium update and retrieval time (Iacob, 2011).

3. Case study. Implementing E-Learning Platform in an Academic Environment using Synchronous Partial Replication

The **E-learning** is a technology that has revolutionized the traditional system of distance learning, and its opportunities have been used not only by educational institutions, but also by different public or private organizations. In terms of structure, an E-learning system provides facilities for the transfer of knowledge through the development and publication of educational content in the form of courses or virtual libraries, as well as testing of knowledge using simulations, scenarios or case studies for evaluation (Iacob, 2011). E-learning facilitates and enhances learning through the use of devices based on computer and communications technology (Defta, 2011).

E-learning is the interaction between teaching and learning, information technology and communications, covering a wide spectrum of activities from computer-aided education to education conducted entirely in an online manner.

E-learning courses include both content (information) and instructional methods (techniques) that help people learn the content (Clark & Mayer, 2008). E-learning type courses are increasingly present in the offer of various universities or academies. The distributed databases are suitable as a technical solution when there are several branches of a geographically dispersed university.

The E-learning portals include a section corresponding to the secretary, mainly with informational purpose, and private sections related to the participants in the E-learning process such as: the private section of a teacher and the private section of a student, accessible to students or teachers registered as users in the electronic platform (a user is an individual recognized by the application based on a username and password for access).

The users of the e-learning platform can access custom web pages depending on the group they belong, by which they can enable:

- administrative procedures by which each user defines its own activity context in training process;
- procedures for adding educational content to the platform (teaching materials, tests, video tutorials);
- procedures for adding information regarding educational process (calendar of activities, advertisements, reminders of events);
- procedures that support the educational process by synchronous contact (videoconferences) and asynchronous contact (discussion forums, e-mails) between teachers and students;
- search procedures in virtual library;
- quality assurance procedures regarding the educational process (questionnaires, statistics).

The application's general objective is to provide logical support (software), in the procedural and functional plan, for the activities of an institution with geographically distributed locations using distributed databases. This is a significant performance optimization, because data are located on the site with “greatest demand” by using data fragmentation and assures a modular growth, because it’s easy to add new systems in this configuration and also offer improved data availability by using data replication technique.

The application consists of an E-learning portal built on Oracle Application Express that meets the growth and expansion challenges of a university with geographically distributed locations. The portal is based on three databases which are named intuitively, based on the city in which they are located: “Bucharest” (usersb), “Timisoara” (userst) and “Paris” (usersp), with fields: *id, name, first_name, personal_id, typeuser, city*.

Using synchronous replication, users from Paris will be replicated from Bucharest in the database of Paris. Table usersb from Paris is called from Bucharest by public synonym usersp, where usersp is usersb@paris_dblink. A

synonym allows access to a table on a remote database using the same syntax that we would use to access a table on a local database. The syntax of the command used for creating table usersb in Oracle is described below:

```
CREATE TABLE usersb
(
ID NUMBER(9,0) NOT NULL ENABLE,
NAME VARCHAR2(20),
FIRST_NAME  VARCHAR2(20),
PERSONAL_ID  NUMBER(20),
TYPEUSERS VARCHAR2(10),
CITY  VARCHAR2(20) CONSTRAINT USERB_PK PRIMARY KEY (ID) ENABLE
)
```

In order to achieve replication between the sites triggers were created for insert, update or delete operations. Triggers are procedures that are stored in the database and are implicitly run when a certain event occurs. Traditionally, triggers support the execution of a PL/SQL block when an INSERT, UPDATE, or DELETE occurred on a table or view. A view is the representation of a SQL statement that is stored in memory so that it can easily be re-used. Thus, when data that is changed in table usersb from Bucharest has the value in column city Timisoara or Paris, the modifications will be replicated to Timisoara or Paris database by using the defined triggers. Below is an example for the implemented UPDATE trigger:

```
CREATE TRIGGER trg_replicationusers_ut
   BEFORE UPDATE ON usersb
      FOR EACH ROW
        WHEN (new.city <> 'BUCHAREST') - we don't consider the data that do not replicate
    BEGIN
      IF (new.city = 'TIMISOARA') THEN
         update userst tim
                set tim.name = new.name, tim.first_name=new.first_name, tim.personal_id = new.personal_id,
                  tim.typeuser=new.typeuser, tim.city = new.city
         where tim.id = new.id;
       ELSIF (new.city = 'PARIS') THEN
          update usersp par
                set par.name = new.name, par.first_name=new.first_name, par.personal_id = new.personal_id,
                  par.typeuser=new.typeuser, par.city = new.city
         where par.id = new.id;
      END if;
    END;
```

In order to achieve primary query optimization, an index was created on a column (the city column in our case):

```
CREATE index CITY_IDX on USERSB (city);
```

4. Conclusion

The replication is a solution for a distributed database environment when we need: to copy and distribute the data to one or more places; to distribute the copies of the data on a scheduled basis; to build applications with data which can be used in online or offline environments; to build Web applications in which the users can access large volumes of data.

The use of distributed databases in E-Learning systems improves access to information and offer rapid data collection.

References

1. R. Clark & R. Mayer, E-learning and the science of instruction. (2nd ed.). San Francisco: John Wiley and Sons, 2008.
2. C.J. Date, An introduction to Database Systems (8th ed.), Addison Wesley, 2003.
3. L. Defta, Information security in E-learning Platforms, Procedia Social and Behavioral Sciences Journal, Volume 15, pp. 2689-2693.
4. N.M. Iacob (Ciobanu), Data Distribution Techniques in Distributed Databases, Annals of Ovidius University, Constanţa, Economic Sciences Series - Volume XI, Issue 1 /2011, pp. 959-962.
5. N.M. Iacob (Ciobanu), Fragmentation and Data Allocation in the Distributed Environments, Annals of the University of Craiova, Mathematics and Computer Science Series, Vol 38, No 3 (2011), pp. 76-83.
6. N.M. Iacob (Ciobanu), The use of distributed databases in e-learning systems, Procedia Social and Behavioral Sciences Journal, Volume 15 (2011), pp. 2673-2677.
7. M.T. Ozsu & P. Valduriez, Principles of Distributed Database Systems. (3th ed.), New York: Springer, 2011.
8. S. Rahimi & F.S. Haug, Distributed database management systems. A Practical Approach, IEEE, Computer society, Hoboken, N. J: Wiley, 2010.
9. A. Silberschatz, H.F. Korth & S. Sudarshan, Database System Concepts. (6th ed.). McGraw-Hill, 2010.